указывает на корректность программной реализации использованного алгоритма.

Литература

1. *Rapaport D. C.* The art of molecular dynamics simulation. Cambridge, 2004.
2. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990.

РАЗРАБОТКА ПРОГРАММ ДЛЯ GRID

А. Г. Лукошко

В настоящее время огромную популярность приобретают GRID-сети. GRID – это географически распределенная инфраструктура, объединяющая множество ресурсов разных типов (процессоры, долговременная и оперативная память, хранилища и базы данных, сети), доступ к которым пользователь может получить из любой точки, независимо от места их расположения. GRID предполагает коллективный разделяемый режим доступа к ресурсам и к связанным с ними услугам в рамках глобально распределенных виртуальных организаций, состоящих из предприятий и отдельных специалистов, совместно использующих общие ресурсы. В каждой виртуальной организации имеется своя собственная политика поведения ее участников, которые должны соблюдать установленные правила. Виртуальная организация может образовываться динамически и иметь ограниченное время существования [1].

Существенная разница в организации сети потребовала пересмотра подхода к программированию. Большое количество новых сервисов и библиотек функций со своими специфическими API требуют длительного освоения.

В данном случае для разработки распределенных приложений было бы предпочтительнее использовать опыт программистов, которые уже разрабатывали параллельные программы. В настоящее время основным стандартом программирования параллельных программ является MPI (Message Passing Interface).

И возможность разрабатывать приложения для GRID по стандарту MPI существует. Специальная реализация MPICH-G2, работающая в среде Globus Toolkit, позволяет компилировать и запускать MPI-приложения без модификации кода.

MPICH-G2 – это реализация Message Passing Interface (MPI) с поддержкой GRID, которая использует сервисы Globus Toolkit (например, запуск задания, обеспечение безопасности) и позволяет программисту соединить много компьютеров, в общем случае, различной архитектуры для выполнения приложений MPI. MPICH-G2 автоматически преобразо-

вывает данные в сообщениях между компьютерами различной архитектуры и поддерживает многопротокольную связь, автоматически выбирая Transmission Control Protocol (TCP) для межмашинной передачи сообщений и поставляемый продавцом MPI протокол для внутримашинной передачи сообщений. Он же освобождает пользователя от продолжительной работы по изучению подробностей, специфичных для конкретных машин, и позволяет пользователю запускать многомашинное приложение одной командой mpirun. MPICH-G2 требует, однако, чтобы сервисы Globus Toolkit были доступны на всех участвующих компьютерах, чтобы можно было войти в контакт с каждой удаленной машиной, подтвердить подлинность пользователя на каждой из них и начать выполнение (например, выполнив команду ветвления fork, поставив задачу в очереди, и т.д.) [2].

Globus Toolkit – инструментарий, разработанный американскими учеными, который стал де-факто мировым стандартом. Он включает в себя, в частности, специальный протокол на основе HTTP для использования вычислительных ресурсов GRAM (Grid Resource Allocation Management); расширенную версию протокола для передачи файлов GridFTP; службу безопасности GSI (Grid Security Infrastructure); распределенный доступ к информации на основе протокола LDAP; удаленный доступ к данным через интерфейс GASS (Globus Access to Secondary Storage) [3].

Схема запуска задания представлена на рисунке 1.

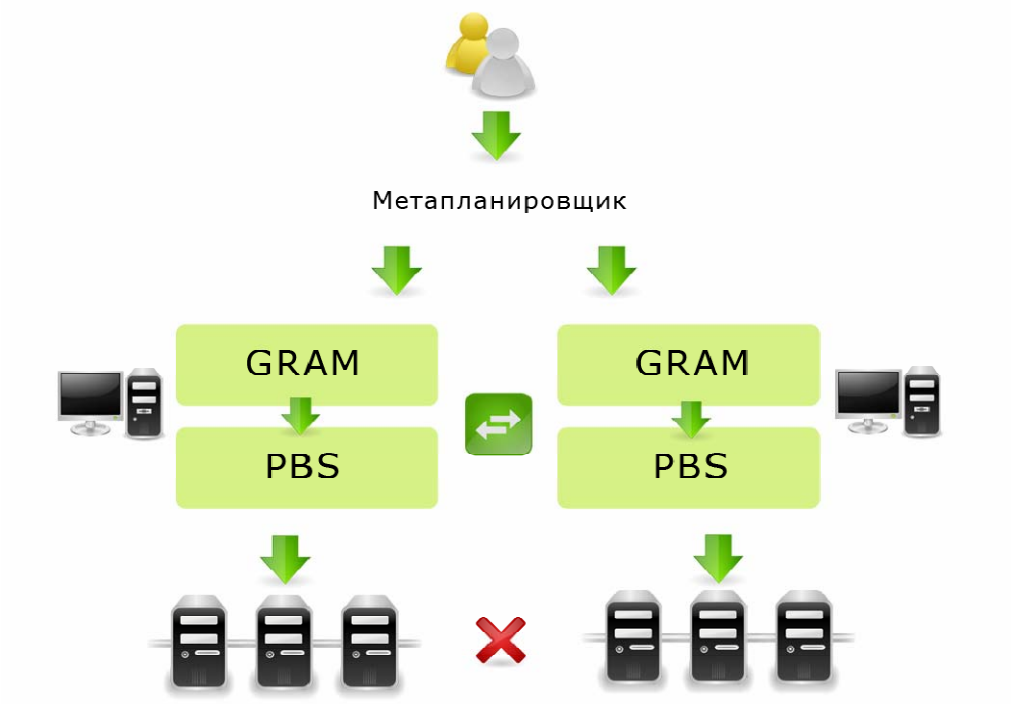

Рис. 1 – схема отправки задания на выполнение

```
+
( &(resourceManagerContact="gridserver1")
(count=20)
(label="subjob 0")
(environment=(GLOBUS_DUROC_SUBJOB_INDEX 0))
(directory="/bin")
(executable="/bin/hpl")
)
( &(resourceManagerContact="gridserver2")
(count=20)
(label="subjob 1")
(environment=(GLOBUS_DUROC_SUBJOB_INDEX 1))
(directory="/bin")
(executable="/bin/hpl")
)
```

Рис. 2. Файл задания для Globus Toolkit
gridserver1, gridserver2 – адреса головных машин; count – количество узлов; environment – переменные окружения; executable – файл запускаемой программы

Метапланировщик выбирает необходимые для выполнения задачи кластеры на основании следующих параметров, предоставленных пользователем:

- Архитектура (i386, Sparc, PowerPC)
- Количество узлов для расчета задачи

По умолчанию метапланировщик старается запустить задание на одном кластере, чтобы избежать межкластерных обменов, которые значительно медленнее, чем межузловые. Если же это невозможно, то он может задействовать 2 и более кластера для вычисления одной задачи. При этом обмены между узлами разных кластеров будут проходить через соответствующие головные машины.

GRAM (GRID Resource Allocation Manager) – осуществляет стандартизованный обмен информацией между метапланировщиком и локальным планировщиком.

Отправка задания происходит традиционным для MPI образом:

mpirun –np N <путь к исполняемому файлу>

Но при этом вместо непосредственно запуска задания происходит промежуточная генерация файла задания для Globus Toolkit.

В данном примере задача будет запущена на двух кластерах одновременно, на задачу будет выделено по 20 узлов на каждом. Результат вычисления будет выведен в консоль пользователю традиционным для MPI образом.

Таким образом, для разработки приложений для GRID вовсе не обязательно осваивать особенности протоколов и служб. Достаточно иметь опыт программирования обычных программ стандарта MPI. Но стоит также заметить, что в таком случае вы воспользуемся далеко не всем перечнем возможностей, которые предоставляют GRID-сети. В случае, ес-

ли это станет необходимо, возможно создание комбинированных приложений.

Литература

1. Интернет-адрес: http://www.gridclub.ru
2. *Karonis N., Toonen B., Foster I.* MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface // Journal of Parallel and Distributed Computing (JPDC). May 2003. V. 63. No. 5. P. 551-563.
3. Интернет-адрес: http://www.globus.org

РАЗРАБОТКА СИСТЕМЫ КОНТРОЛЯ ПОСЕЩЕНИЯ НА ОСНОВЕ ТЕХНОЛОГИИ BLUETOOTH

А. Э. Мазго, П. В. Петров, Д. В. Бобров, Н. Н. Кольчевский

Современное образовательное учреждение активно внедряет новые информационные технологии в своем учебном процессе. Использование компьютера, компактных информационных носителей, сети Интернет помогают расширить сферу образовательных услуг и радиус их действий. На сегодняшний день в Белгосуниверситете развитие программы дистанционного образования реализуется на основе систем: eUniversity, содержащей электронные обучающие средства, медиатеки, предоставляющие возможность использовать компьютер, компактные информационные носителей, сеть Интернет. Дистанционное образование базируется на функциях электронной почты, т.е. предоставлении электронных учебных материалов и контрольных работ посредством электронной почтовой переписки. Использование современных компьютеризированных технологий позволяет также изменить процедуры проведения лабораторных работ и лекций. Такие технологии базируются на определенном уровне аппаратных решений. Необходимо компьютерное оборудование, программное обеспечение, наличие выхода в Интернет, наличие сервера и т.д.

Целью данной работы является построение системы мониторинга, использующейся во время лекционных занятий. Система мониторинга подразумевает организацию обратной связи со студентами во время чтения лекции. Для этой цели предлагается использовать компьютер с установленным радио-модулем Bluetooth и мобильные телефоны, оснащенные радио-модулями Bluetooth. В качестве задач предложены следующие: проверка присутствующих студентов на занятии, проверка степени усвояемости материала, рассылка тестовых заданий, прием решений тестовых заданий. Данные задачи предполагают решение следующих технических задач: обнаружение устройств и радио-модулей Bluetooth; рассылка и прием пакетов по найденным устройствам; разработка тестовых